

1 **METHOD FOR EXECUTING MULTI-SYSTEM AWARE APPLICATIONS**

2 **Technical Field**

3 The present invention relates to system administration management, and, in
4 particular, to ServiceControl Manager modules.

5 **Background**

6 Computer systems are increasingly becoming commonplace in homes and
7 businesses throughout the world. As the number of computer systems increases, more
8 and more computer systems are becoming interconnected via networks. These
9 networks include local area networks (LANs). LANs also frequently have an interface
10 to other networks, such as the Internet, and this interface needs to be monitored and
11 controlled by network management on the LAN.

12 One concern encountered with networks is referred to as network management.
13 Network management refers to monitoring and controlling of the network devices and
14 includes the ability for an individual, typically referred to as an administrative user, to
15 be able to access, monitor, and control the devices that are part of the network, or
16 access, monitor, and control the devices that are part of the network coupled to other
17 computer systems. Such access, monitoring, and control often include the ability to
18 check the operating status of devices, receive error information for devices, change
19 configuration values, and perform other management functions. As the size of
20 networks increases, so too does the need for network management.

21 The operating system of most computers provides an administration tool or a
22 system administration manager for invoking and performing system management tasks.
23 The hardware of a computer system, the various facilities included within the operating
24 system, such as the file system facility, the print spooling facility, and the networking
25 facility, as well as the operating system itself must all be managed. This means that
26 computer systems require some involvement by a human user or a manager of the
27 computer system for such operations as specifying certain configuration parameters,
28 monitoring ongoing activity, or troubleshooting some problem that has arisen. These
29 management or administration tasks can be performed manually in many operating
30 systems, for example, by a risk control manager, via direct manipulation of

1 configuration files or direct invocation of specific administration utility programs. But
2 in most modern operating systems, an easy to use, interactive software program is
3 typically provided that hides the details of the file formats and the utility program
4 syntax, while providing a higher level presentation for the system administrator.

5 The System Administration Manager (SAM) is a system manager used by
6 Hewlett-Packard Co's version (HP-UX) for Unix system administration management.
7 Contemporary versions of SAM include a graphical user interface (GUI) arranged to
8 make as user-friendly as possible the various system administration activities that are
9 available with SAM. In a conventional Unix system, the user performing such
10 administration activities must be a root user, referred to as a super-user. The super-user
11 has unlimited privileges with regard to the reading and writing of files, and with regard
12 to what commands he or she may execute in the system. SAM allows the super-user to
13 perform the most important tasks in a system administrator's job by filling out
14 templates, rather than using command line interfaces. SAM allows system
15 administrators to perform basic administrative tasks, such as adding new users,
16 installing new printers, assigning administration privileges, and reconfiguring a kernel
17 with a new print driver, quickly, easily and, most importantly, safely.

18 Another system manager, referred to as a Software Distributor (SD), is the HP-
19 UX administration tool-set used to deliver and maintain HP-UX operating systems and
20 layered software applications. SD allows central IT departments to control an
21 associated software environment. It also improves administrator productivity by
22 automating software distribution.

23 SAM management applications can only operate on the system on which the
24 applications are running, while SD/UX applications have the ability to operate on
25 multiple nodes. HP's OpenView Network Node Manager provides a simple means to
26 integrate single-system aware (SSA) applications such as SAM, but a completely
27 different and more complex means for integrating multi-system aware (MSA)
28 applications such as SD/UX. For example, since Open View can only run a tool on a
29 remote machine as root, integrating SD/UX into OpenView Network Node Manager

1 requires a complex process for implementing a number of changes by a whole project
2 team.

3 There are other similar distributed management systems such as IBM's Tivoli
4 product that behave similarly to OpenView by providing a very complex mechanism for
5 integrating multi-system aware (MSA) management applications. There is a need for a
6 simple mechanism to integrate SSA applications and MSA applications.

7 **Summary**

8 A ServiceControl Manager (SCM) module provides a simple means to integrate
9 both single system aware (SSA) management applications and multi-system aware
10 (MSA) management applications into a single multi-system management environment.

11 MSA applications may be started by a user using either command line interface
12 (CLI) or graphical user interface (GUI). Using GUI, there may be two different ways of
13 executing MSA applications on a remote node, either from a tool view menu or from a
14 node view menu.

15 Either from CLI or from GUI, the method for MSA applications includes
16 selecting an MSA tool by a user, establishing a target node list that contains nodes on
17 which the tool may run, and passing the target node list as environment variables. The
18 environment variables are then passed to the MSA applications that use the node list to
19 restrict the user access to these nodes.

20 **Description of the Drawings**

21 The detailed description refers to the following drawings, in which like numbers
22 refer to like elements, and in which:

23 Figure 1(a) illustrates a computer network system with which the present
24 invention may be used;

25 Figures 1(b) and 1(c) compare single-system aware tools and multi-system
26 aware tools;

27 Figure 2 illustrates the relationships between the user, role, node, tool and
28 authorization objects;

29 Figure 3(a) is a block diagram of an exemplary server used to implement the
30 present invention;

1 Figures 3(b) and 3(c) shows a tool view menu and a node view menu,
2 respectively;

3 Figure 4 illustrates a method for executing MSA applications in the SCM
4 module using a command line interface;

5 Figure 5 illustrates a method for executing MSA applications using a graphical
6 user interface from a tool view menu; and

7 Figure 6 illustrates a method for executing MSA applications using a graphical
8 user interface from a node view menu.

9 **Detailed Description**

10 A ServiceControl Manager (SCM) module multiplies system administration
11 effectiveness by distributing the effects of existing tools efficiently across managed
12 servers. The phrase "ServiceControl Manager" is intended as a label only, and different
13 labels can be used to describe modules or other entities having the same or similar
14 functions.

15 In the SCM domain, the managed servers (systems) are referred to as "managed
16 nodes" or simply as "nodes". SCM node groups are collections of nodes in the SCM
17 module. They may have overlapping memberships, such that a single node may be a
18 member of more than one group.

19 Figure 1(a) illustrates a computer network system with which the present
20 invention may be used. The network system includes an SCM 110 running on a Central
21 Management Server (CMS) 100 and one or more nodes 130 or node groups 132
22 managed by the SCM 110. The one or more nodes 130 and node groups 132 make up
23 an SCM cluster 140. For a more detailed description of an embodiment of SCM, see
24 ServiceControl Manager Technical Reference, HP® part number: B8339-90019,
25 available from Hewlett-Packard Company, Palo Alto, CA., which is incorporated herein
26 by reference and which is also accessible at
27 <<http://www.software.hp.com/products/scmgr>>.

28 The CMS 100 can be implemented with, for example, an HP-UX 11.x server
29 running the SCM 110 software. The CMS 100 includes a memory 102, a secondary
30 storage device (not shown), a processor 108, an input device (not shown), a display

1 device (not shown), and an output device (not shown). The memory 102 may include
2 computer readable media, RAM or similar types of memory, and it may store one or
3 more applications for execution by processor 108, including the SCM 110 software.
4 The secondary storage device may include computer readable media, a hard disk drive,
5 floppy disk drive, CD-ROM drive, or other types of non-volatile data storage. The
6 processor 108 executes the SCM software and other application(s), which are stored in
7 memory or secondary storage, or received from the Internet or other network 116. The
8 input device may include any device for entering data into the CMS 100, such as a
9 keyboard, key pad, cursor-control device, touch-screen (possibly with a stylus), or
10 microphone. The display device may include any type of device for presenting a visual
11 image, such as, for example, a computer monitor, flat-screen display, or display panel.
12 The output device may include any type of device for presenting data in hard copy
13 format, such as a printer, and other types of output devices include speakers or any
14 device for providing data in audio form. The CMS 100 can possibly include multiple
15 input devices, output devices, and display devices.

16 The CMS 100 itself may be required to be a managed node, so that multi-system
17 aware (MSA) tools may be invoked on the CMS. All other nodes 130 may need to be
18 explicitly added to the SCM cluster 140. Alternatively, the CMS 100 may be part of the
19 SCM cluster 140.

20 Generally, the SCM 110 supports managing a single SCM cluster 140 from a
21 single CMS 100. All tasks performed on the SCM cluster 140 are initiated on the CMS
22 100 either directly or remotely, for example, by reaching the CMS 100 via a web
23 connection 114. Therefore, the workstation 120 at which a user sits only needs a web
24 connection 114 over a network 116, such as the Internet or other type of computer
25 network, to the CMS 100 in order to perform tasks on the SCM cluster 140. The CMS
26 100 preferably also includes a centralized data repository 104 for the SCM cluster 140,
27 a web server 112 that allows web access to the SCM 110 and a depot 106 that includes
28 products used in the configuring of nodes 130. A user interface may only run on the
29 CMS 100, and no other node 130 in the SCM module may execute remote tasks, access
30 the repository 104, or any other SCM operations.

1 Although the CMS 100 is depicted with various components, one skilled in the
2 art will appreciate that this server can contain additional or different components. In
3 addition, although aspects of an implementation consistent with the present invention
4 are described as being stored in memory, one skilled in the art will appreciate that
5 these aspects can also be stored on or read from other types of computer program
6 products or computer-readable media, such as secondary storage devices, including hard
7 disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other network; or
8 other forms of RAM or ROM. The computer-readable media may include instructions
9 for controlling the CMS 100 to perform a particular method.

10 A central part of the SCM module 110 is the ability to execute various
11 management commands or applications on the one or more nodes simultaneously. The
12 commands or applications may need to be encapsulated with an SCM tool, which is
13 typically used to copy files and/or execute commands on the target nodes 130. The
14 SCM tool may run simple commands such as bdf (1) or mount (1M), launch single
15 system interactive applications such as System Administration Manager (SAM) or
16 Glance, launch multi-system aware applications such as Ignite/UX or Software
17 Distributor (SD), or perform other functions. The tool may be defined using either an
18 SCM tool definition language through command line interface (CLI) or an SCM-
19 provided graphical user interface (GUI).

20 There are two general types of tools: single-system aware (SSA) tools and multi-
21 system aware (MSA) tools. SSA tools, illustrated in Figure 1(b), may run on a node
22 130 and may only affect the operation of that node 130. To run SSA tools on multiple
23 target nodes 130, the SCM module 110 may execute the tools on each target node 130.
24 In addition to executing commands or launching applications, SSA tools may copy files
25 from the CMS 100 to the target nodes 130. Files may only be copied from the CMS
26 100 to the managed nodes 130 in this exemplary embodiment, not from the nodes 130
27 back to the CMS 100.

28 MSA tools, illustrated in Figure 1(c), may run on a single node 130 but may be
29 able to operate on multiple other nodes 130. MSA tools are applications that execute on
30 a single node but can detect and contact other nodes to accomplish their work and this

1 contact is out of the control of the SCM module 110. This type of application may need
2 to have a list of nodes 130 passed as an argument at runtime. A node 130 where the
3 application will execute may need to be specified at tool creation time, not at runtime.
4 The target nodes 130 selected by the user may be passed to an MSA tool via
5 MX_TARGETS environment variables (described later). MSA tools may not copy files
6 to either the manager node 100 or to the target nodes 130 in this exemplary
7 embodiment. Therefore, an execution command string may be required for MSA tools.

8 An SCM user may be a user that is known to the SCM module 110 and has
9 some privileges and/or management roles. An SCM role, which is an expression of
10 intent and a collection of tools for accomplishing that intent, typically defines what the
11 user is able to do on the associated nodes 130 or node groups 132, e.g., whether a user
12 may run a tool on a node 130. Typically, in order to start the SCM module 110 or
13 execute any SCM tools, the user may need to be added to the SCM module 110 and
14 authorized either via the GUI or the command line interface (CLI). All SCM module
15 110 operations may be authorized based on the user's SCM authorization configuration,
16 and/or whether or not the user has been granted SCM trusted user privilege.

17 The SCM user may, depending upon the roles assigned, manage systems via the
18 SCM module 110. In addition, the user may examine an SCM log, and scan the group
19 and role configurations. When the SCM user runs a tool, the result may be an SCM
20 task. The SCM module 110 typically assigns a task identifier for every task after it has
21 been defined and before it is run on any target nodes 130. This identifier may be used
22 to track the task and to look up information later about the task in an SCM central log.

23 An SCM trusted user is an SCM user responsible for the configuration and
24 general administration of the SCM cluster 140. The trusted user is typically a manager
25 or a supervisor of a group of administrators whom a company trusts, or other trusted
26 individual. The capabilities of the trusted user include, for example, one or more of the
27 following: creating or modifying a user's security profile; adding, modifying or deleting
28 a node or node group; tool modification; and tool authorization. The granting of these
29 privileges implies a trust that the user is responsible for configuring and maintaining the
30 overall structure of the SCM module 110.

1 An SCM authorization model supports the notion of assigning to users the
2 ability to run a set of tools on a set of nodes. An authorization object is an association
3 that links a user to a role on either a node or a node group. Each tool may belong to one
4 or more roles. When users are given the authority to perform some limited set of
5 functionality on one or more nodes, the authorization is done based upon roles and not
6 on tools. The role allows the sum total of functionality represented by all the tools to be
7 divided into logical sets that correspond to the responsibilities that would be given to
8 the various administrators. Accordingly, there are different roles that may be
9 configured and assigned with authorization. For example, a backup administrator with
10 a "backup" role may contain tools that perform backups, manage scheduled backups,
11 view backup status, and other backup functions. On the other hand, a database
12 administrator with a "database" role may have a different set of tools. When a user
13 attempts to run a tool on a node, the user may need to be checked to determine if the
14 user is authorized to fulfill a certain role on the node and if that tool contains the role.
15 Once a user is assigned an authorization, the user gains access to any newly created
16 tools that contain the role. In the example given above, the backup administrator may
17 be assigned the "backup" role for a group of systems that run a specific application.
18 When new backup tools are created and added to the "backup" role, the backup
19 administrator immediately gains access to the new tools on the systems.

20 Figure 2 illustrates the relationships between user 210, role 220, node 130, tool
21 240, and authorization 250 objects. User objects 210 represent users 210, role objects
22 220 represent roles 220, node objects 130 represent nodes 130, tool objects 240
23 represent tools 240, and authorization objects 250 represent authorizations 250.
24 However, for the purpose of this application, these terms are used interchangeably.
25 Each authorization object 250 links a single user object 210 to a single role object 220
26 and to a single node object 130 (or a node group object 132). Each role object 220 may
27 correspond to zero or more tool objects 240, and each tool object 240 may correspond
28 to one or more role objects 220. Each user object 210 may be assigned multiple
29 authorizations 250, as may each role object 220 and each node object 130. For
30 example, Role 1 may contain Tools 1-N, and User 1 may be assigned Roles 1-M by the

1 authorization model on Node 1. Consequently, User 1 may run Tools 1-N on Node 1,
2 based upon the role assigned, Role 1.

3 Table 1 illustrates an example of a data structure for assigning tools 240 and
4 commands specified in the tools 240 to different roles 220. Table 2 illustrates an
5 example of a data structure for assigning the roles 220 to different users 210 on
6 different nodes 130.

Roles	Tools	Commands and Applications
Role 1	Tools 1-N	Commands 1-L
.....
Role n	Tools 1-Nn	Commands 1-Ln

8
9 Table 1
10

Users	Assigned Roles	Assigned Nodes
User 1	Roles 1-M	Nodes 1-x
.....
User n	Roles 1-M	Nodes 1-x

11
12 Table 2
13

14 Although Figure 2 shows a node authorization, a similar structure exists for a
15 node group 132 authorization. The SCM authorization model may be deployed by
16 using node group 132 authorizations more often than node 130 authorizations. This
17 model makes adding new nodes simpler because by adding a node 130 to an existing
18 group 132, any authorizations associated with the group 132 may be inherited at run-
19 time by the node 130.

1 The authorization model for determining if a user may execute a tool 240 on a
2 set of nodes 130 may be defined by an “all or none” model. Therefore, the user 210
3 must have a valid authorization association for each target node 130 to execute the tool
4 240. If authorization does not exist for even one of the nodes 130, the tool execution
5 fails.

6 The SCM cluster 140 may also include security features to secure transactions
7 that transmit across the network. All network transactions may be digitally signed using
8 a public or private key pair. The recipient of network transmissions may be assured of
9 who the transmission came from and that the data was not altered in the transmission.
10 A hostile party on the network may be able view the transactions, but may not
11 counterfeit or alter them.

12 Referring to Figure 3(a), the CMS 100 may include a domain manager 330, a
13 log manager 334, and a distributed task facility (DTF) 240. The domain manager 330 is
14 the “brain” of SCM module 110 and may be connected to the repository 104 for storage
15 of the definitions of all the objects.

16 The DTF 340 may execute tasks by passing the task definitions and information
17 to agents running on the managed nodes 130. The DTF 340 is the “heart” of all task
18 execution activity in that all of the execution steps must go through the DTF 340. The
19 DTF 340 typically obtains an authorized runnable tool from a client, distributes the tool
20 execution across multiple nodes 130, and returns execution results to the clients and to
21 the user 210.

22 An integral part of the SCM functionality may be the ability to record and
23 maintain a history of events, by logging both SCM configuration changes and task
24 execution events through the log manager 334. The log manager 334 may manage a log
25 file and take log requests from the DTF 340, the graphical user interface, and the
26 command line interface, and write the requests to the SCM log file. SCM configuration
27 changes may include adding, modifying and deleting users and nodes in the SCM
28 module 110, and creating, modifying and deleting node groups 132 and tools 240. An
29 example of task execution events may include details and intermediate events
30 associated with the running of a tool 240. An example of task execution is described in

1 United States patent application of Lister, Sanchez, Drees, and Finz, Serial No.
2 09/813,562, entitled "Service Control Manager Tool Execution", and filed on March 20,
3 2001, which is incorporated herein by reference. The details that are logged may
4 include the identity of the user 210 who launched the task, the actual tool and command
5 line with arguments, and the list of target nodes 130. The intermediate events that are
6 logged may include the beginning of a task on a managed node 130, and exceptions that
7 occur in attempting to run a tool 240 on a node 130, and the final result, if any, of the
8 task. The exit code, standard output (stdout) and standard error output (stderr), if exist,
9 may also be logged.

10 A security manager 332, which is a subsection of the domain manager 330,
11 typically guards the system security by checking whether the user 210 is authorized to
12 run the tool 240 on all of the nodes 130 requested, i.e., whether the user 210 is assigned
13 the roles 220 associated with the tool 240 on all of the nodes 130, and whether the
14 necessary roles 220 are enabled on a particular tool 240.

15 A tool 240 may be started in an SCM environment, which is the memory set
16 aside for the tool 240 to look up attribute values. When launching MSA applications,
17 the SCM environment may be extended to pass additional information by providing
18 additional environment variables. For example, MX_USER is an environment variable
19 that contains the login name of the user 210 executing the tool 240; MX_TASKID is an
20 environment variable that contains the DTF task ID and uniquely names a tool
21 execution instance; MX_TOOL is an environment variable that contains the name of
22 the tool 240 that executed this specific executable; MX_TARGETS is an environment
23 variable that contains the application's target node list for MSA applications, the list of
24 node names may be space-delimited and sorted in a lexicographic order; MX_CMS is
25 an environment variable that contains the host name of the CMS 100; and
26 MX_REPOSITORY is an environment variable that contains the hostname of the
27 system hosting the SCM repository 104. When a user 210 with authorization to nodes
28 1-5 launches a tool 240, the SCM determines an identity of the user and establishes
29 environment variables that contain environment variable value pairs, so that only nodes
30 1-5 can be accessed by this user 210. Accordingly, the behavior of these applications is

1 different when they run stand-alone and when they run in the SCM environment, where
2 they have to follow the rules set by the SCM. If the user 210 tries to access resources
3 outside that domain, the attempt will be blocked by the MSA tool 240 and an error
4 message returned.

5 Applications may be integrated into the SCM environment by creating an SCM
6 tool 240 for them. This tool 240 may have a wrapper script that may process any input
7 parameters and run the application. The application software may need to be pre-
8 installed on the target nodes 130. Some applications may be distributed by nature and
9 may be encapsulated in the distributed tool 240. When a task is executing on a node
10 130, the agent running on the node 130 may set the environment variables in the
11 environment in which the tool command runs.

12 In a GUI, SCM integrated applications may be categorized, for example, based
13 on the shade of blue that an application turns as it uses more and more of the SCM
14 functionality. A deeper shade of blue may indicate more and more use of the SCM
15 functionality. By implication, the darker shaded applications use most if not all of the
16 functionality of a lighter shade application. Table 3 describes each shade.

Clear	These applications use none of the SCM functionality.
Light Blue	These applications are moderately SCM aware, in that during installation they detect the presence of a CMS. As part of their configuration they provide information to the SCM module and expose their set of tools to the SCM.
True Blue	These applications are not only SCM aware during installation, they are aware that when launched they can take advantage of the additional information passed to them by the SCM module. They may also make use of the SCM authorization model, and the SCM logging facility to ensure that their own application logging is integrated with the SCM and thus highly searchable. Likewise, True Blue applications adhere to the SCM GUI look and feel.
Deep Blue	These are the most highly integrated applications, that use the SCM data repository not only to acquire SCM specific information but also to store

their own application specific information.

Table 3

Light Blue distributed applications may build command lines using the target environment variable.

True Blue applications are aware that when launched they are provided with additional start up information via the environment variables. Furthermore, True Blue applications may integrate their application logging into the SCM central log file, which may provide the added benefit of centralized logging. Likewise, end users 210 may take advantage of the SCM log file querying mechanism. Additionally, True Blue applications may take advantage of the SCM roles 220 by running application specific tools 240 under the guise of the user 210 identified in the MX_USER environment variable, assuming that the True Blue applications initiate their actions from the CMS 100.

Deep Blue applications may be the most tightly integrated because they have knowledge of the SCM data repository 104, its schema, and the directory structure. Deep Blue applications may read SCM user 210 and node 130 entries. They may extend the data repository 104 by complying with the directory layout structure and the schema extension rules, and storing their application specific data in their portion of the data repository 104.

SSA applications may only run on the node 130 that the applications reside. The procedure to launch SSA applications is similar to the procedure to launch MSA applications, which is described next.

MSA applications (tools) may run at the CMS 100 or an MSA managed node 130 other than the CMS 100. To launch the MSA applications and pass a specific list of remote target nodes 130, default targets may be established during tool definition to specify the target nodes 130 on which to the MSA applications should affect configuration changes. Alternatively, the target nodes 130 on which to execute the MSA applications may be selected by the user 210, in which case, the default targets may be ignored by the DTF 340. An example of tool definition is described in United

1 States patent application of Lister, Sanchez, Drees, and Finz, Serial No. 09/800,316,
2 entitled "Service Control Manager Tool Definition", and filed on March 6, 2001, which
3 is incorporated herein by reference.

4 Tasks may be started by a user 210 using either the CLI or GUI. Using the GUI,
5 there may be two different ways of executing tasks on a remote node 130, either from a
6 tool view menu or from a node view menu. The tool view menu 360, shown in Figure
7 3(b), provides a view of the tools 240 defined in the SCM cluster 140. The uppermost
8 view is of tool categories 350, which are container objects, containing tools 240. Tools
9 240 may be assigned to a category 350 when created. When opened, the tools 240 in
10 that category 350 may be displayed, and the user 210 can double-click on a tool 240 to
11 execute it. The nodes view menu 370, shown in Figure 3(c), provides a view of the
12 SCM nodes 130 from which to initiate actions. A trusted user can see all the nodes 130
13 in the cluster 140, while other users 210 can only see and select the nodes on which they
14 have authorizations. Figure 4 illustrates a method for executing MSA applications in
15 the SCM cluster 140 using the CLI, Figure 5 illustrates a method for executing MSA
16 applications using GUI from the tool view menu 360, and Figure 6 illustrates a method
17 for executing MSA applications using GUI from the node view menu 370. These
18 methods may be implemented in, for example, software modules for execution by
19 processor 108.

20 Referring to Figure 4, from the CLI, a task may be started manually by typing a
21 command, such as mxexec. The user then identifies on the mxexec command line an
22 MSA tool 240 to run in the SCM environment, step 410. If the user 210 has specified
23 the target nodes 130 to run the tool 240 on the command line, step 420, the mxexec CLI
24 in the CMS 100 may establish a target node list that contains the user specified target
25 nodes 130, and ignore any tool specified default targets, step 450. On the other hand, if
26 the user 210 fails to specify the target nodes 130, the mxexec CLI may check whether
27 there are default targets 130 specified, step 430. If yes, from the default targets 130, the
28 mxexec CLI may compute a default target list that contains the nodes the user 210 is
29 able to access, step 440, and a target node list may be established from the default node
30 list, step 450. However, if there are no target nodes specified by the user 210, and no

1 default targets 130 specified, the mxexec command line may return an error message to
2 the user 210, such as “need to specify a target”, step 480. After the target node list is
3 established, the DTF 340 may pass the target node list as the MX_TARGETS
4 environment variable, step 460. Now, the selected target or default target nodes 130
5 become the contents of the environment variable that may later be passed to the MSA
6 tool 240. The MSA tool 240 may then be executed on the MSA managed node 130,
7 and emanate to the nodes 130 in the target node list, step 470.

8 Finally, since SCM cluster configuration changes may be logged by the log
9 manager 334 in an SCM central log file, while tool execution events may be logged in
10 an MSA tool log file, the MSA tool log file may be integrated into the SCM central log
11 file to provide the added benefit of centralized logging, step 490. Likewise, end users
12 210 may take advantage of the SCM log file querying mechanism.

13 As mentioned above, running an MSA tool in the SCM environment using the
14 GUI may start from, for example, either the tool view menu 360 or the node view menu
15 370, the steps of which are described in Figures 5 and 6, respectively. Referring to
16 Figure 5, from the GUI tool view menu 360, a user 210 first selects an MSA tool 240,
17 step 510. Next, the GUI checks whether there are default targets 130 specified in the
18 tool definition, step 520. If yes, from the default targets 130, the GUI may compute a
19 default target list that contains the nodes the user 210 is able to access, step 530, and a
20 target node list may be established from the default node list, step 560. Conversely, if
21 there are no default targets 130 specified, the user 210 may be presented with a “run tool
22 dialog”, step 540, to manually select the target nodes 130 on which to run the tool 240,
23 step 550. Then, similar to establishing a target node list from the default targets, the
24 GUI may establish a target node list that contains the target nodes 130 selected by the
25 user 210, step 560. After the target node list is established, the DTF 340 may pass the
26 target node list as the MX_TARGET environment variable, step 570, so that the default
27 target or the selected target nodes 130 become the contents of this environment variable
28 that may later be passed to the MSA tool 240. The MSA tool 240 may then be executed
29 on the MSA managed node 130, and emanate to the nodes 130 in the target node list,

1 step 470. Finally, the MSA tool log file may be integrated into the SCM central log file
2 to provide added benefit of centralized logging, step 490.

3 From the node view menu 370, as processed by the method shown in Figure 6, a
4 user 210 first selects the desired target nodes 130 on which to run the tool 240, step
5 610. Then the user 210 is taken to a "run tool dialog" by selecting, for example, a "user
6 choose tool" bar, step 620. From the "run tool dialog", the user 210 may select an MSA
7 tool 240, step 630, and the GUI in the CMS 100 may establish a target node list that
8 contains the selected target nodes 130 from which the tool 240 may allow access to the
9 user 210, step 640, and pass the target node list as the MX_TARGETS environment
10 variable, step 650. Now, the selected target nodes 130 become the contents of this
11 environment variable that may be passed to the MSA tool 240. The MSA tool 240 may
12 then be executed on the MSA managed node 130, and emanate to the nodes 130 in the
13 target node list, step 470. Finally, the MSA tool log file may be integrated into the
14 SCM central log file to provide added benefit of centralized logging, step 490.

15 In summary, by setting up a single multi-system management environment, the
16 SCM module 110 provides a simple method to integrate both SSA management
17 applications and MSA management applications into the multi-system environment.

18 While the present invention has been described in connection with an exemplary
19 embodiment, it will be understood that many modifications will be readily apparent to
20 those skilled in the art, and this application is intended to cover any variations thereof.